REMARKS

I. INTRODUCTION

Claims 1-7 remain pending in the present application. In view of the following remarks, it is respectfully submitted that all of the presently pending claims are allowable.

II. THE 35 U.S.C. § 102(e) REJECTION SHOULD BE WITHDRAWN

Claims 1-7 stand rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,308,255 to Gorishek IV et al. ("Gorishek"). (See 07/13/07 Office Action, p. 3, ¶ 4).

Claim 1 recites, a "method, comprising: *determining a current processing mode of an executing software function*; when the current processing mode is a privileged processing mode, executing a direct program flow control instruction to directly access an instruction within a software having the privileged processing mode; and *when the current processing mode is an unprivileged processing mode*, executing an indirect program flow control instruction to cause execution of the instruction within the software having the privileged processing mode." (Emphasis added).

Gorishek generally relates to a computer system including a host processor and an emulation coprocessor. (See Gorishek, Abstract). According to Gorishek, the "host processor" comprises hardware configured to execute instructions defined by a host instruction set architecture, while the "emulation coprocessor" comprises hardware configured to execute instructions defined by a different (or foreign) instruction set architecture. (See Id., col. 6, lines 28). Accordingly, the host processor executes operating system code as well as application programs that are coded in the host instruction set architecture, while the emulation coprocessor executes the foreign application program. (See Id.). Therefore, when a user submits a command

to initiate an application program, the system examines the file format of the application program in order to determine what type of code is included in the application program. (See Id., col. 13, lines 44-49). If the application program is determined to be coded according to the host instruction set architecture, the system establishes the process as a normal host process and the application is executed by the host processor. (See Id., col. 13, lines 44-49). Alternatively, if the application program is determined to be coded according to a foreign instruction set architecture executable by the emulation coprocessor, then the system invokes the emulation coprocessor interface code in order to initiate the foreign application program. (See Id., col. 13, line 55 – col. 14, line 2). Thus, the initiation of an application program according to the system and method disclosed by Gorishek is dependent on the application program, specifically the code within the application program. In other words, the system according to Gorishek determines which of the two processors (the host or the emulation coprocessor) will execute the code of an application program depending on the format of the code.

The Examiner asserts that the determining by Gorishek of the type of code included in the application program is equivalent to the determining a current processing mode of an executing software function, as recited in claim 1 of the present invention. (See 07/13/06 Office Action, p. 3, ¶ 4). However, this assertion is incorrect. The determination of a processing mode, as recited in claim 1 of the present invention, is not made on an application program basis and the processing mode is not dependant on the type of code that is included within the application program. Specifically, the determination made in the Gorishek method examines the code of the application program to be executed, while the determination, as recited in claim 1, examines "a current processing mode of an executing software function." In contrast to the Gorishek disclosure, a privileged processing mode may allow for the execution of a particular

application program, while an unprivileged processing mode may deny the execution of that very same application program. Since the system and method disclosed by Gorishek is dependent on the code of the application program requested by the user, the host processor will execute *any* application program that includes code formatted for the host processor. Likewise, the emulation coprocessor will execute *any* application program that includes code according to its foreign instruction set architecture. However, there is no determination made as to the current processing mode of either the host processor or the emulation coprocessor. Thus, it is respectfully submitted that Gorishek does not disclose nor suggest, "determining a current processing mode of an executing software function" as recited in claim 1.

Accordingly, Applicants respectfully submit that for at least the reasons stated above, claim 1 of the present application is not anticipated by Gorishek, and request that the rejection of this claim be withdrawn. As claims 2-5 depend from, and therefore include all the limitations of claim 1, it is hereby submitted that these claims are also allowable.

Claim 6 recites, *inter alia*, "identify a current processing mode of the program code segment" and "execute an indirect program flow control instruction if the current processing mode is an unprivileged processing mode." Thus, for the reasons described above with reference to claim 1, it is respectfully submitted that claim 6 is also allowable.

Claim 7 recites, *inter alia*, "identify a current processing mode of a program code segment" and "execute an indirect program flow control instruction if the current processing mode is an unprivileged processing mode." Thus, for the reasons described above with reference to claim 1, it is respectfully submitted that claim 7 is also allowable.

CONCLUSION

In light of the foregoing, Applicants respectfully submit that all of the now pending claims are in condition for allowance. All issues raised by the Examiner having been addressed. An early and favorable action on the merits is earnestly solicited.

Respectfully submitted,

Dated: October 15, 2007

By: _____
Michael J. Marcin (Reg. No. 48,198)

Fay Kaplun & Marcin, LLP
150 Broadway, Suite 702
New York, NY 10038
Phone: 212-619-6000
Fax: 212-619-0276